Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Edge Detection Using Ant Colony Search Algorithm and Multiscale Contrast Enhancement

Aleksandar Jevtić, Joel Quintanilla-Dominguez, M.G. Cortina-Januchs and Diego Andina
Group for Automation in Signals and Communications
Technical University of Madrid (UPM)
Madrid, Spain
aleksandar.jevtic@upm.es, {joelq, januchs}@salamanca.ugto.mx, d.andina@upm.es

***Abstract*—In this paper, Ant Colony System (ACS) algorithm is applied for edge detection in grayscale images. The novelty of the proposed method is to extract a set of images from the original grayscale image using Multiscale Adaptive Gain for image contrast enhancement and then apply the ACS algorithm to detect the edges on each of the extracted images. The resulting set of images represents the pheromone trails matrices which are summed to produce the output image. The image contrast enhancement makes ACS algorithm more effective when accumulating pheromone trails on the true edge pixels. The results of the experiments are presented to confirm the effectiveness of the proposed method.**

***Index Terms*—Ant colony system, Edge detection, Nonlinear image enhancement.**

I. Introduction

Swarm Intelligence systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The group of individuals acting in such a manner is referred to as a swarm [1]. The term stigmergy is used to describe the indirect form of communication between individuals in a swarm via environment. Individuals within the group interact by exchanging locally available information such that the global objective is reached more efficiently than it would be done by a single individual.

Ant Colony System (ACS) is a swarm intelligence algorithm which exploits the self-organizing nature of real ant colonies in their search for food [2]. Ant-based algorithms have been used to successfully solve many complex problems, such as the travelling salesman problem [3], quadratic assignment problem [4], data mining [5], data clustering [6], and image retrieval [7]. Ant-based algorithms have been as well proposed to solve the problem of edge detection [8], [9].

Edge detection is a preprocessing step in applications of image segmentation and computer vision. Edges represent important contour features in the corresponding image as they are the boundaries where distinct intensity changes or discontinuities occur. Edge detection is a process which transforms a grayscale image to a binary image which indicates either the presence or the absence of an edge [10], [11]. In practice, it is impossible to design an edge detector capable of finding all true, and only the true edges in an image. Furthermore, edge detectors give ambiguous information about location of object boundaries. This usually results in edge detectors being subjectively evaluated by observers [12].

Numerous edge detection techniques have been proposed. The Prewitt operator was proposed to extract contour features by fitting a Least Squares Error (LSE) quadratic surface over a 3×3 image window and differentiate the fitted surface [13]. The edge detectors such as Sobel [14] or Canny [15] use local gradient operators, sometimes with additional smoothing for noise removal. The Laplacian operator uses a second order differential operator to find edge points based on the zero crossing properties of the processed edge points [16]. Although conventional edge detectors usually perform linear filtering operations, there are various nonlinear methods proposed. Panetta *et al.* in [17] proposed a logarithmic edge detection method based on Parameterized Logarithmic Image Processing (PLIP) and a four directional Sobel method, achieving a higher level of independence from scene illumination. Xiangjian *et al.* [18] presented an edge detection method based on bilateral filtering which achieves better performance than single Gaussian filtering. Mertzios and Tsirikolias [19] proposed using the Coordinate Logic Filters (CLF) for purpose of edge extraction. CLF constitute a class of nonlinear digital filters that are based on the execution of Coordinate Logic Operations (CLO). Danahy *et al.* [20] introduced an alternative method for calculating CLF using Coordinate Logic Transforms (CLT) and presented a new measure and thresholding technique for the detection of edges in grayscale images.

In this paper, we propose a method that combines nonlinear image enhancement with ACS-based edge detector. Multiscale Adaptive Gain as a nonlinear image enhancement technique is described in Section II. Section III presents the ACS algorithm where the distributed system of agents (ants) and their self-organizing nature are described as a result of an iterative process. The proposed edge detection method as the combination of the Multiscale Adaptive Gain and ACS algorithm is described in Section IV. Experimental results are also presented in this section. Finally, in Section V conclusions are made.

II. Nonlinear Image Enhancement

Image enhancement techniques emphasize important features in image while reducing the noise. In order to accomplish multiscale contrast enhancement, Multiscale Adaptive Gain is applied by suppressing pixel values of very small amplitude and enhancing only those pixels larger than a certain threshold within each level of the transform space. The nonlinear

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

operation is described with the following equation [21]:

$$G(I) = \alpha[sigm(k(I - B)) - sigm(-k(I + B))] \quad (1)$$

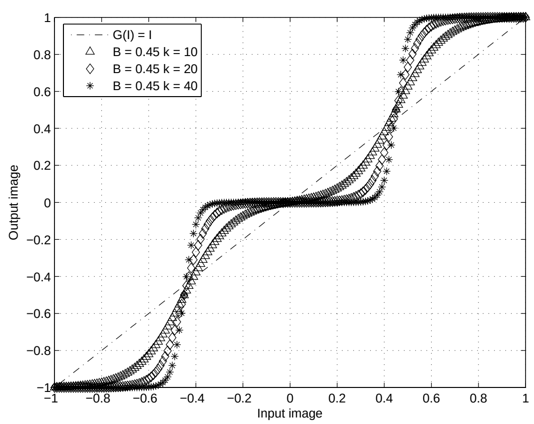
where

$$\alpha = \frac{1}{sigm(k(1 - B)) - sigm(-k(1 + B))} \quad (2)$$

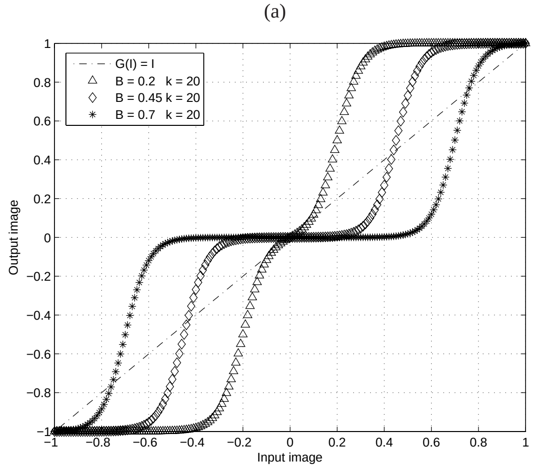
where $I = I(i, j)$ is the gray value of the pixel at (i, j) of the input image and $sigm(x)$ is defined as

$$sigm(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

and B and k control the threshold and rate of enhancement, respectively. ($0 < B < 1$, $B \in \Re$; $k \in \aleph$). The transformation function (1) in respect to the original image pixel values is shown in Fig. 1. It can be seen that $G(I)$ is continuous and monotonically increasing, therefore, the enhancement will not introduce new discontinuities into the reconstructed image.



(a)



(b)

Fig. 1. Transformation function $G(I)$ in respect to the original image pixel values: (a) B = 0.45; k = 10, 20 and 40; (b) B = 0.2, 0.45 and 0.7; k = 20.

III. Ant Colony System Algorithm

ACS algorithm is based on the foraging behavior of ant colonies. The artificial ants, unlike their biological counterparts, move through a discrete environment defined with nodes, and they have memory. When moving from one node to another, artificial ant leaves a pheromone trail on its route. The pheromone trail attracts other ants, which by positive feedback leads to a pheromone trail accumulation. Negative feedback is applied through the pheromone evaporation which, importantly, restrains the ants from taking the same route, i.e. prevents the algorithm stagnation.

After defining the discrete environment in which the artificial ants can move, the ACS algorithm starts with an initialization step which is followed by iterative construction of new solutions and pheromone update. The ACS algorithm involves the following steps:

1) Initialization: certain number of ants is placed on randomly chosen nodes.
2) Node transition rule: ants probabilistically determine the next node to move to. The probability of displacing kth ant from node i to node j is given by:

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{h \notin tabu_k,} (\tau_{ih})^\alpha (\eta_{ih})^\beta} & \text{if j} \quad \notin \quad \text{tabu}_k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where τ_{ij} and η_{ij} are the intensity of the pheromone trail on edge (i, j) and the visibility of the node j from node i, respectively, and α and β are control parameters ($\alpha, \beta > 0$; $\alpha, \beta \in \Re$). Tabu$_k$ list contains the nodes that have already been visited by the kth ant.
3) Pheromone update rule: after every ant has constructed a solution of which pixel to move to, the pheromone update is applied:

$$\tau_{ij,(new)} = (1 - \rho)\tau_{ij,(old)} + \sum_{k=1}^{m} \Delta\tau_{ij}^k \quad (5)$$

where ρ is the pheromone evaporation rate ($0 < \rho < 1$, $\rho \in \Re$), and $\Delta\tau_{ij}^k$ is the amount of pheromone laid on edge (i, j) by the kth ant, and is given by:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{f_k}{Q} & \text{if edge } (i,j) \text{ is traversed by the } \\ & \text{k}th \text{ ant at the current cycle} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where f_k is the fitness value of the solution found by kth ant and Q is a constant.
4) Stopping criterion: The steps 2 and 3 are repeated in a loop and algorithm stops executing when the acceptable solution is found or the maximum number of iterations is reached.

IV. The Proposed Method and Experimental Results

The proposed method is based on the foraging behavior of ant colonies where, in search for food, ants leave pheromone trails in order to attract other ants to follow their route. For the edge detection problem, the routes are the edges of the objects in image and pixels in the image represent the nodes.

The input to the ACS algorithm is a set of images generated from the original grayscale image by applying the Multiscale Adaptive Gain as a nonlinear image contrast enhancement. The resulting set of images are the matrices of pheromone trails accumulated on the edge pixels. The output image is obtained as a sum of the pheromone trails matrices. A global threshold is applied to discard the irrelevant edges, and morphological thinning gives the final result. The block diagram of the proposed method is shown in Fig. 2.

A. Multiscale Adaptive Gain

The adaptive histogram enhancement defined in (1) is applied to the input image: $0 \leq I(i,j) \leq I_{max}$, i = 1,2,...,N; j = 1,2,...,M. The values of B and k are varied to obtain the set of 9 enhanced images: B = 0.2, 0.45 and 0.7; k = 10, 20 and 40. The effects of the transformation function (1) on the input image "cameraman" are shown in Fig. 3.

B. Edge Detection Based on Ant Colony System Algorithm

The ACS algorithm is applied to each of the 9 enhanced images. It is an iterative process of Nc cycles with Ni iterations, and includes the following steps:

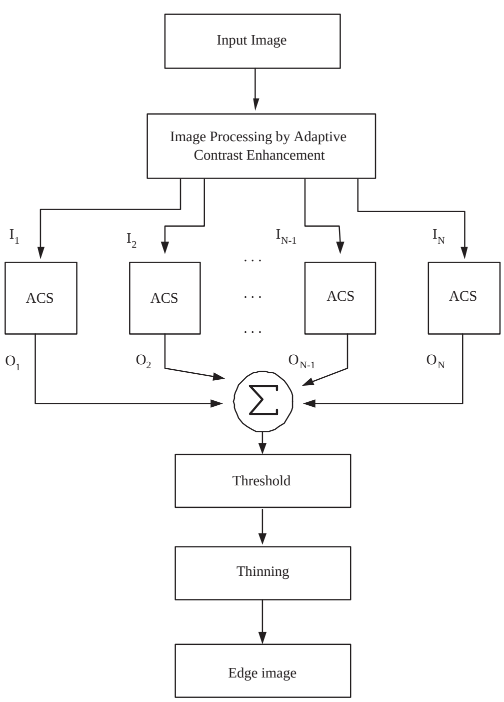


Fig. 2. Block diagram of the proposed edge detection method

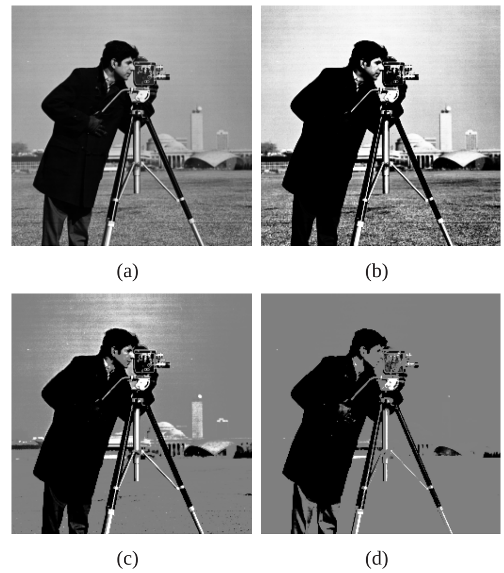


Fig. 3. Effects of the transformation function $G(I)$; "Cameraman" 256×256 pixels: (a) Original image; (b) B = 0.2, k = 10; (c) B = 0.45, k = 20; (d) B = 0.7, k = 40.

1) Initialization: the number of ants proportional to $\sqrt{N \cdot M}$ is randomly distributed on the pixels in the image. Only one ant is allowed to reside on a pixel within the same iteration. Initial pheromone trail for each pixel is set to 0.01, otherwise by setting it to zero, the ants would never move to a neighboring pixel. The visibility of the pixel at (i,j) is calculated as follows:

$$\eta_{ij} = \frac{1}{I_{max}} \cdot max \begin{bmatrix} |I(i-1,j-1) - I(i+1,j-1)|, \\ |I(i-1,j+1) - I(i+1,j+1)|, \\ |I(i,j-1) - I(i,j+1)|, \\ |I(i-1,j) - I(i+1,j)| \end{bmatrix} \tag{7}$$

where I_{max} is the maximum gray value in the image. For the pixels in regions of distinct gray intensity changes the higher visibility values are obtained.

2) Pixel transition rule: If the kth ant is found on a non-edge pixel or surrounded by the pixels that are either in the tabu$k$ list either occupied by another ants, it is randomly placed to another pixel that is neither occupied by another ant neither found in the tabu$_k$ list. Otherwise, the probability for the kth ant to move from pixel (r,s) to pixel (i,j) is calculated by:

$$p^k_{(r,s)(i,j)} = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_u \sum_v (\tau_{uv})^\alpha (\eta_{uv})^\beta} & \text{if}(i,j) \text{ and } (u,v) \text{ are allowed nodes } r-1 \leq i,u \leq r+1, \; s-1 \leq j,v \leq s+1 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where τ_{ij} and η_{ij} are the intensity of the pheromone trail and the visibility of the pixel at (i, j), respectively, and α and β are control parameters ($\alpha, \beta > 0$; $\alpha, \beta \in \Re$). Designated values α = 2.5 and β = 2 were determined on the trial and error basis. The ant memory, i.e. the length of the tabu list was set to 10. Possible ant's transitions to the neighboring pixels are shown in Fig 4.

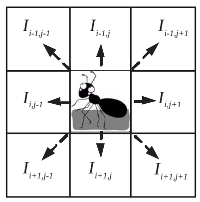


Fig. 4. Proposed pixel transition model

3) Pheromone update rule: Negative feedback is demonstrated through the pheromone trails evaporation according to:

$$\tau_{ij} = (1-\rho)\tau_{ij}(old) + \Delta\tau_{ij} \tag{9}$$

where

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}, \tag{10}$$

and

$$\Delta\tau_{ij}^{k} = \begin{cases} \eta_{ij} & \text{if } \eta_{ij} \geq T \text{ and } kth \text{ ant displaces to pixel } (i,j) \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

T is a threshold value. Pheromone evaporation prevents algorithm stagnation. The value ρ = 0.04 used in the experiments was chosen by trial and error. In case of repeatedly not-visited pixels pheromone trail evaporates exponentially.

4) Stopping criterion: The steps 2 and 3 are repeated in a loop and algorithm stops executing when the maximum number of cycles and iterations is reached. In our experiments, the values that gave satisfactory results for the acceptable computational time of execution were: the number of cycles was chosen to be 3, and the number of iterations was set to 40 and 80, for 128×128 and 256×256 pixel images, respectively.

The effectiveness of the proposed algorithm is best presented by the results shown in Fig. 5. The method was tested on the image "Cameraman" of two different sizes, 128×128 and 256×256 pixels. The self-organizing behavior of ants can be seen on the pheromone trail images (Fig. 5 (a) and (b)). A global threshold is applied to remove the irrelevant edges, i.e. the pixels with lower accumulation of pheromone trails. Finally, by applying morphological edge-thinning the resulting binary images are obtained (Fig. 5 (c) and (d)).

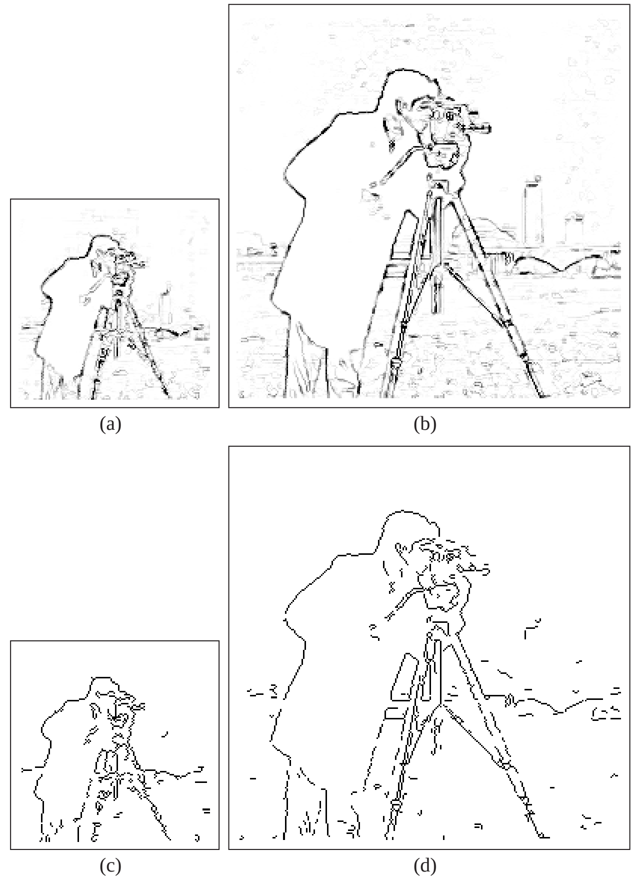


Fig. 5. Qualitative results of the proposed method for "Cameraman" (a) 128×128 pixels pheromone trail image; (b) 256×256 pixels pheromone trail image; (c) 128×128 edge image; (d) 256×256 edge image.

In order to test the proposed method on different input images, experiments were performed on "House", "Lena" and "Peppers" images of size 256×256 pixels. The results of these experiments confirm the universal nature of the method, as shown in Fig. 6. In altered images, ACS algorithm effectively detects the true edges as the self-organizing behavior of the ant colony optimizes the routes defined by edge pixels. The number of cycles and iterations was empirically determined. If it would be increased, somewhat better results could be obtained but it would also increase the computational cost of the algorithm.

V. Conclusions

Edge detection is of utmost importance for image analysis and processing. Swarm Intelligence techniques have shown to be able to successfully tackle with this challenge. In this paper, we presented a method that combined the multiscale contrast enhancement of grayscale images with the Ant Colony System algorithm for edge detection. The excellent performance of the proposed method was confirmed with experimental results. Parameters of the algorithm were empirically determined so the future work could involve their additional optimization.

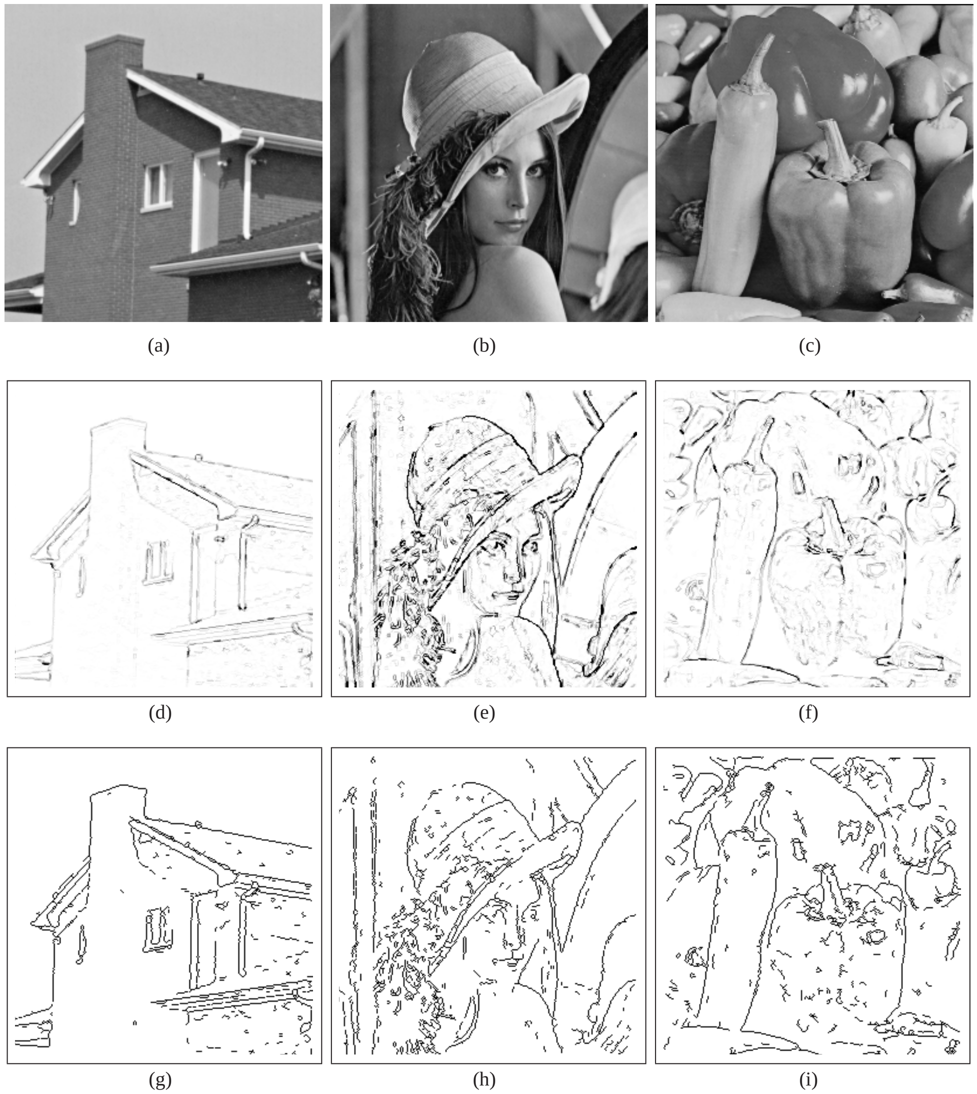


Fig. 6. Qualitative results of the proposed method 256×256 pixel images (a) "House" original image; (b) "Lena" original image; (c) "Peppers" original image; (d) "House" pheromone trail image; (e) "Lena" pheromone trail image (f) "Peppers" pheromone trail image; (g) "House" edge image (h) "Lena" edge image (i) "Peppers" edge image.

REFERENCES

[1] A. Jevtić and D. Andina. Swarm intelligence and its applications in swarm robotics. *Proceedings of the 6th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics 2007 (CIMMACS '07)*, pages 41–46, 2007.

[2] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, 2004.

[3] M. Dorigo and L.M. Gambardella. Ant colony system: a cooperating learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computing*, 1:1–24, 1997.

[4] Dorigo M. Stutzle T. ACO algorithms for the quadratic assignment problem. *New ideas in optimization, McGraw-Hill NewYork*, pages 33–50, 1999.

[5] R.S. Parpinelli, H.S. Lopes, and A.A. Freitas. Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computing*, 6:4, 2002.

[6] Zhongzhi Shi Bin Wu. A clustering algorithm based on swarm intelligence. *International Conferences on Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing.*, 3:58–66, 2001.

[7] V. Ramos, F. Muge, and P. Pina. Self organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies. *2nd International Conference on Hybrid Intelligent Systems*, 87:500–509, 2002.

[8] H. Nezamabadi-Pour, S. Saryazdi, and E. Rashedi. Edge detection using ant algorithms. *Soft Computing*, 10:623–628, 2006.

[9] Jing Tian, Weiyu Yu, and Shengli Xie. An ant colony optimization algorithm for image edge detection. *IEEE Congress on Evolutionary Computation*, pages 751–756, 2008.

[10] J.A. Jiang, C.L. Chuang, Y.L. Lu, and C.S. Fahn. Mathematical-morphology-based edge detectors for detection of thin edges in low-contrast regions. *Image Processing, IET*, 1(3):269–277, September 2007.

[11] T. Chen, Q. H. Wu, R. Rahmani-Torkaman, and J. Hughes. A pseudo top-hat mathematical morphological approach to edge detection in dark regions. *Pattern Recognition*, 35(1):199 – 210, 2002.

[12] M.C. Shin, D. Goldgof, and K.W. Bowyer. Comparison of edge detector performance through use in an object recognition task. *Computer Vision and Image Understanding*, 84(1):160–178, 2001.

[13] J.M.S. Prewitt. *Object Enhancement and Extraction in Picture Processing and Psychopictorics*. Academic Press, New York, 1970.

[14] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing. *presented at a talk at the Stanford Artificial Project, unpublished but often cited*, 1968.

[15] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[16] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2 edition, 2002.

[17] K. A. Panetta, E. J. Wharton, and S.S. Agaian. Logarithmic edge detection with applications. *Journal of Computers (JCP)*, 3(9):11–19, September 2008.

[18] H. Xiangjian, J. Wenjing, H. Namho, W. Qiang K. Jinwoong, and T. Hintz. *Advances in Visual Computing*, chapter Bilateral Edge Detection on a Virtual Hexagonal Structure, pages 176–185. Springer, 2006.

[19] B.G. Mertzios and K. Tsirikolias. Applications of coordinate logic filters in image analysis and pattern recognition. *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, 2001. ISPA 2001.*, pages 125–130, 2001.

[20] E. E. Danahy, K. A. Panetta, and S. S. Agaian. Coordinate logic transforms and their use in the detection of edges within binary and grayscale images. *IEEE International Conference on Image Processing, 2007. ICIP 2007.*, 3:53–56, 2007.

[21] Andrew F. Laine, Sergio Schuler, Jian Fan, and Walter Huda. Mammographic feature enhancement by multiscale analysis. *IEEE Transactions on Medical Imaging*, 13:725–740, 1994.